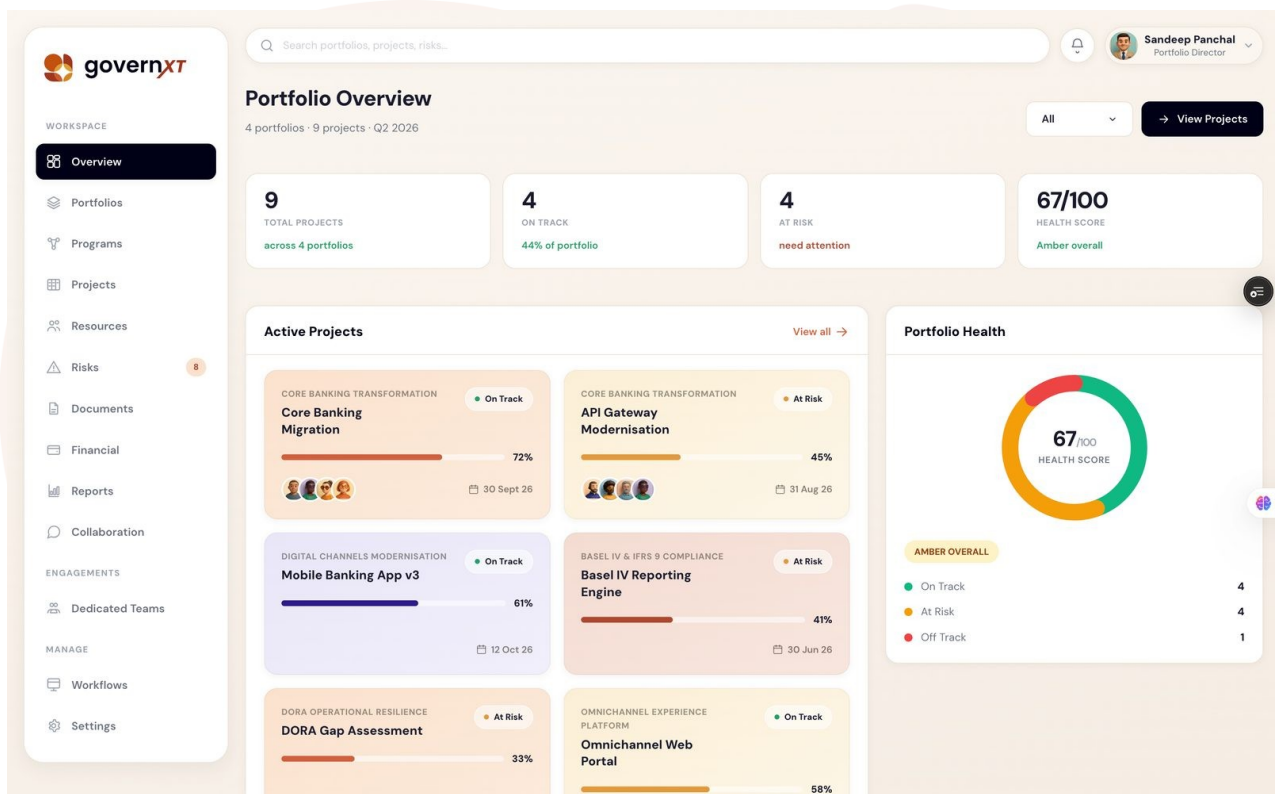




Enterprise Project, Program & Portfolio Governance Platform



The GovernXT Portfolio Overview — live at www.governxt.com

PRODUCT CASE STUDY

Sandeep Panchal · Platform Lead & Sole Developer

July 2026 · www.governxt.com

Executive Summary

GovernXT is a multi-tenant, enterprise-grade Project, Program and Portfolio (3P) governance platform built for large multinational organisations. It unifies portfolio strategy, program delivery, project execution, resource capacity, risk, financial control, change governance and document management into a single, role-aware workspace — deployed globally and running on a modern serverless stack.

The platform was designed, built and shipped end-to-end by a single developer: sixteen live dashboard routes, a three-layer security model, a bespoke design system and a multi-tenant platform administration console, all fetching live data from a row-level-secured Postgres backend. This case study walks through the product, the architecture and the engineering practices behind it. The platform is populated with a realistic banking-sector demo portfolio so every screen shown here renders genuine data from the production database.

At a Glance

| Dimension | Detail |
|--------------|--|
| Live product | www.governxt.com — deployed on Vercel (Mumbai region, bom1) |
| Frontend | Next.js 16 (App Router) · React 19 · TypeScript |
| Backend | Supabase — Postgres, Auth, Storage · Row-Level Security on every table |
| Modules | 11 integrated modules across 16 live dashboard routes |
| Security | 7 system roles · RBAC enforced at 3 independent layers · 21 configurable role labels |
| Data model | 22+ production tables, all org-scoped with RLS · versioned documents · approval chains |
| Design | Bespoke design system — warm palette, custom iconography, AI-generated 3D avatars |
| Team | Designed, engineered and operated by one person |

The Challenge

Enterprise PPM tooling tends to fall into two camps: heavyweight suites that take quarters to configure, or lightweight task trackers that collapse under governance requirements. Large organisations — especially in regulated sectors such as banking — need portfolio-level strategic alignment, program dependency management, earned-value cost control, auditable change approval chains and strict role separation, without sacrificing the modern, fast UX their teams expect.

GovernXT was conceived to prove that a single, coherent product can deliver that full governance stack: C-suite readable at the top, delivery-team practical at the bottom, and secure by construction throughout.

The Solution

A single workspace where every governance artefact — portfolio, program, project, task, risk, change request, document, financial record — lives in one relational model and is filtered through one permission system. Executives see health scores; PMO sees approval queues and capacity heatmaps; project managers see Kanban, Gantt and WBS views — and every user sees exactly what their role permits, enforced in the database itself rather than only in the UI. The remainder of this case study tours each module, then examines the security model, the design system and the engineering practice underneath.

Architecture & Technology

GovernXT runs on a fully serverless architecture with the entire stack region-pinned to Mumbai for latency consistency between compute and data.

| Layer | Technology | Role in the platform |
|---------------|----------------------------------|---|
| Presentation | Next.js 16 App Router · React 19 | Server components, streaming rendering, per-route code-splitting across 16 dashboard routes |
| Language | TypeScript | End-to-end typed contracts between UI, API routes and database rows |
| Data & Auth | Supabase (Postgres 15+) | Relational core, cookie-based session auth, Storage buckets for avatars and documents |
| Authorisation | Postgres Row-Level Security | Org-scoped policies on all 22+ tables; DB triggers guard critical invariants |
| Hosting | Vercel (bom1) | Edge-deployed production builds, preview deployments, function logs |
| Styling | Bespoke CSS design system | Single dashboard.css token system — no UI framework dependency |

Security in Depth

Access control is enforced at three independent layers, so a defect in any one layer cannot expose data:

- **Layer 1** — Frontend route guards
- **Layer 2** — Middleware permission checks on every request, using cookie-read sessions for zero-network validation
- **Layer 3** — Postgres Row-Level Security policies scoped to the user's organisation and role — the database is the final authority

Critical business rules are also guarded inside the database itself; for example, a trigger prevents the last administrator of an organisation from being demoted, regardless of what any client sends.

Product Tour — Eleven Modules

Every screen below is a live production render with role-based data. The tour follows the platform's own navigation: from portfolio strategy down to delivery, then across governance, analytics and administration.

Portfolio Overview

The landing dashboard answers the executive question — how healthy is the portfolio right now? Nine projects across four portfolios roll up into a single 67/100 health score, split into on-track, at-risk and off-track segments. Each project card carries its program, RAG status, progress and deadline, with team avatar stacks rendered by the platform's avatar service.

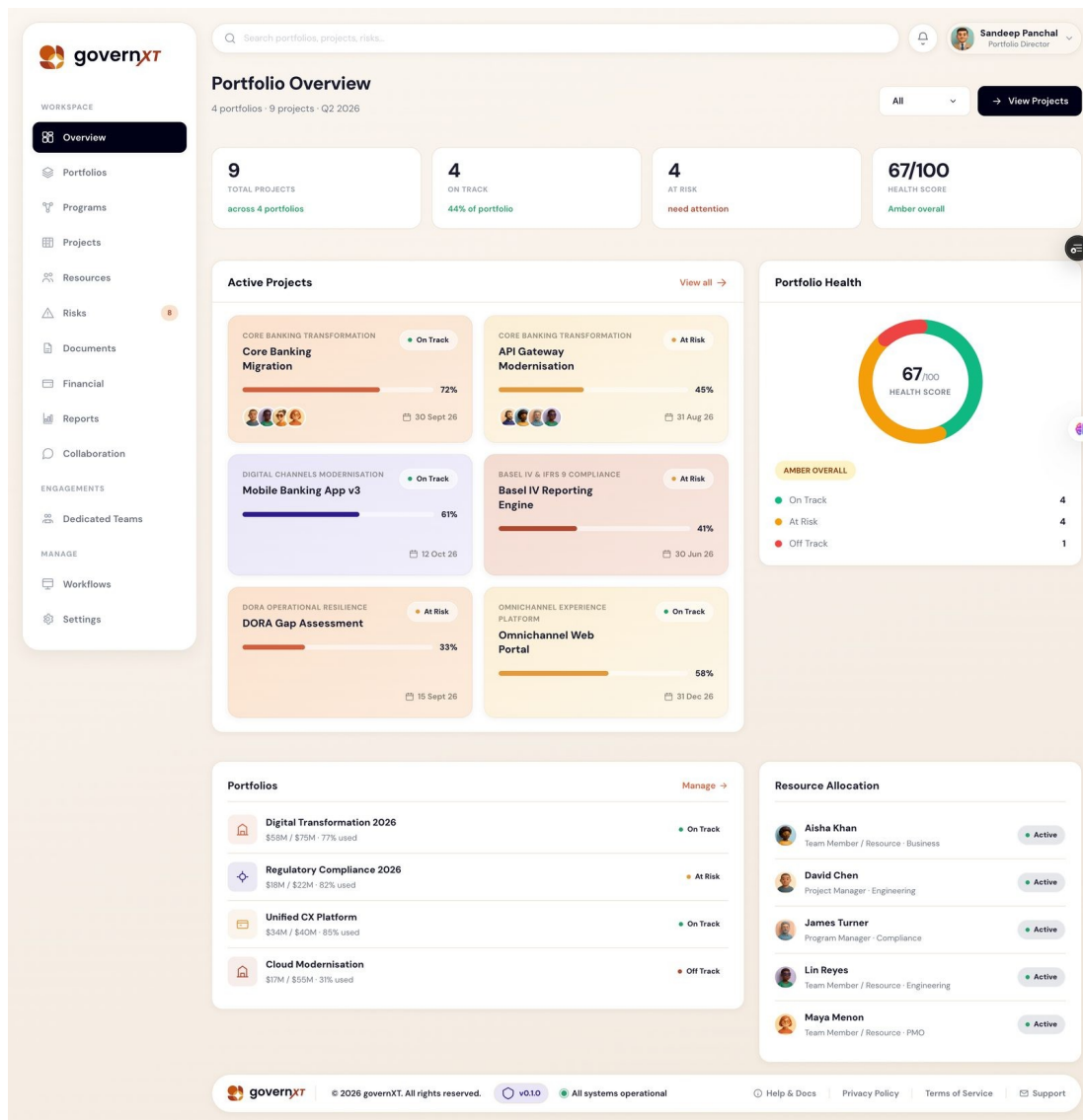


Figure 1 — Portfolio Overview: KPI cards, active projects and portfolio health donut

Portfolio Management

Portfolios are the strategic layer: each card tracks budget consumption against allocation, strategic alignment and benefit realisation. An upcoming-milestones rail keeps go-lives, sign-offs and dry-runs visible. Full CRUD with multi-step creation modals is role-gated to the Portfolio Director.

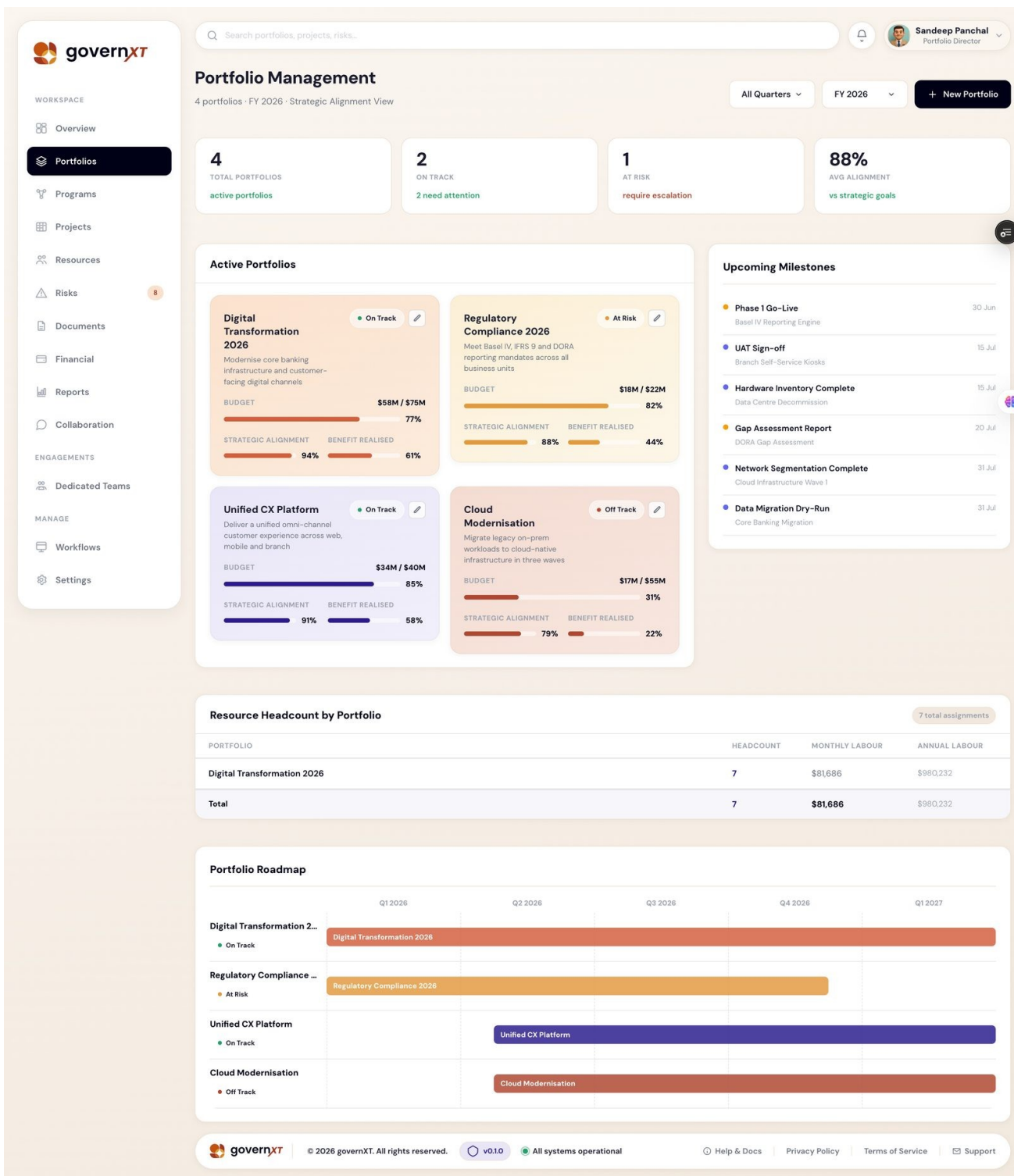


Figure 2 — Portfolio Management: strategic alignment, budget burn and benefit realisation per portfolio

Program Management

Programs group related projects under a shared objective. The module tracks per-program budget, milestone countdowns and — critically for multinationals — cross-program dependencies with finish-to-start relationships and a change log, so a slip in one program is visible to every dependent one.

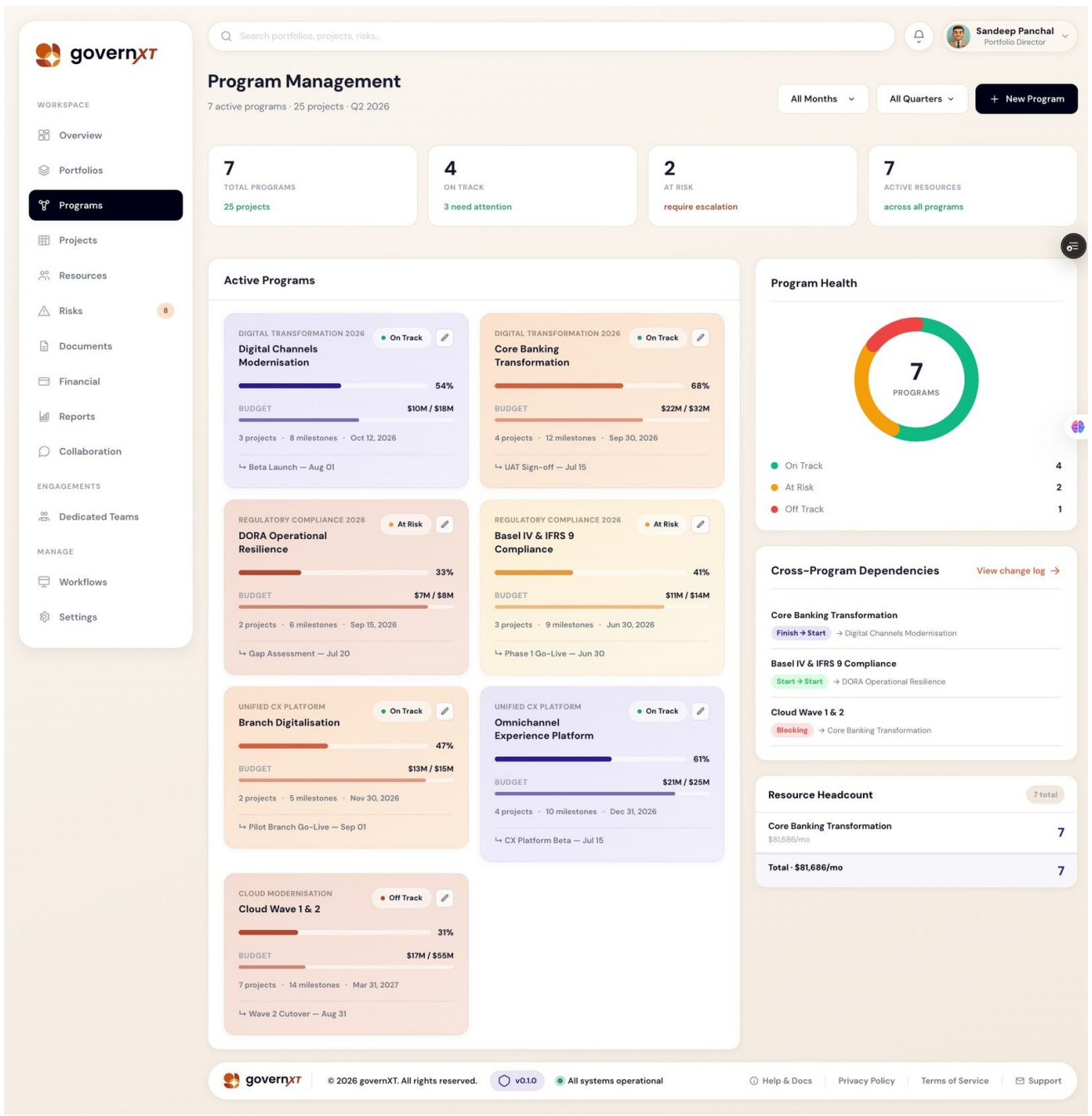


Figure 3 — Program Management: seven programs with milestones, budgets and cross-program dependencies

Project Management

The project ledger provides the delivery view: nine projects with lifecycle stage, RAG status, progress, budget burn and due dates, with overdue flags computed live. Behind the list view sit Kanban boards with configurable columns, Gantt timelines and a drag-to-reorder WBS tree — all wired to a live tasks table.

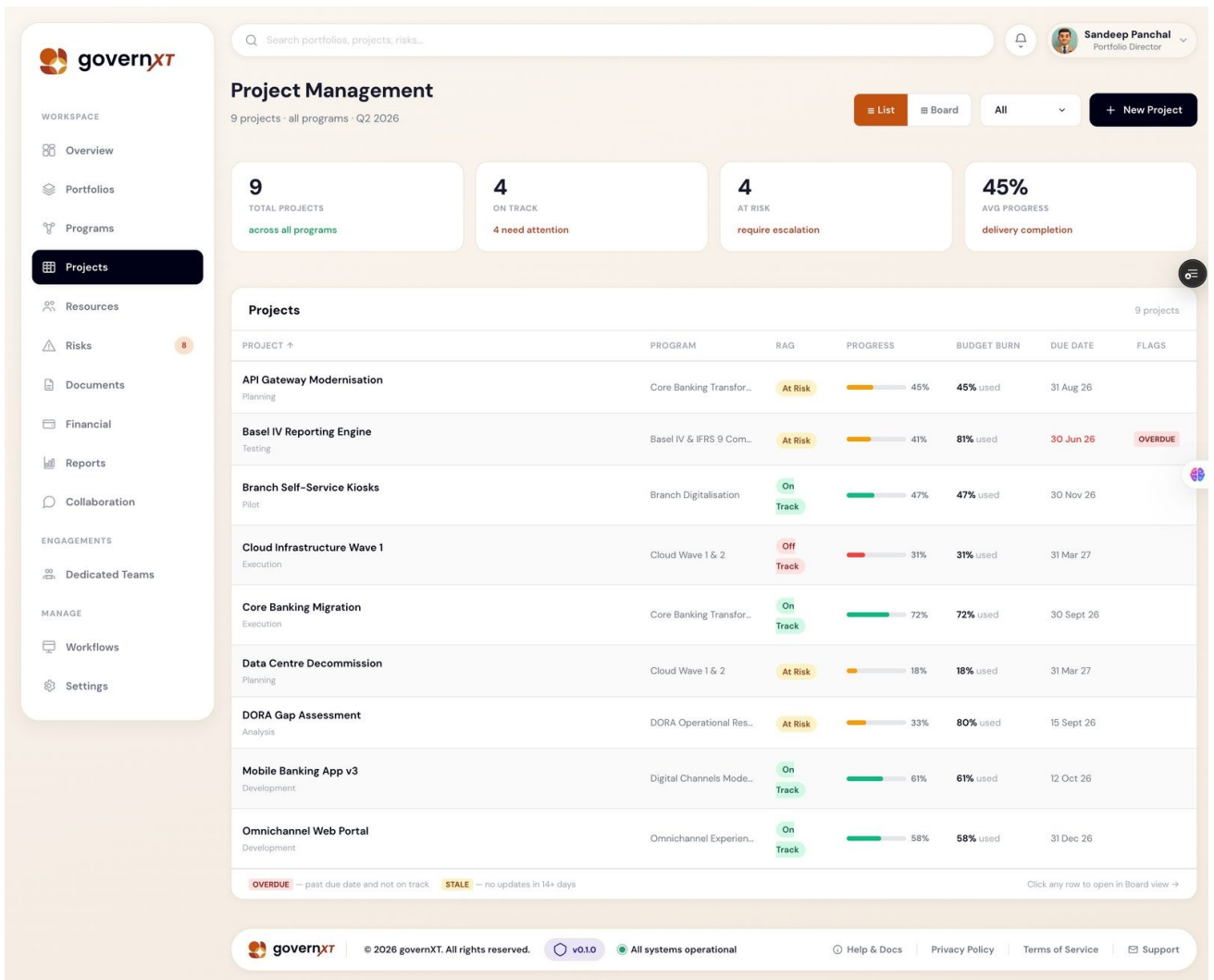


Figure 4 — Project Management: sortable delivery ledger with RAG, budget burn and overdue flags

Resource Management

Capacity is rendered as an eight-week heatmap per resource, colour-graded from under-utilised to over-committed, with warning markers on over-allocated people. Team allocation cards carry skills, day rates, engagement type and utilisation; a department rollup summarises load per function.

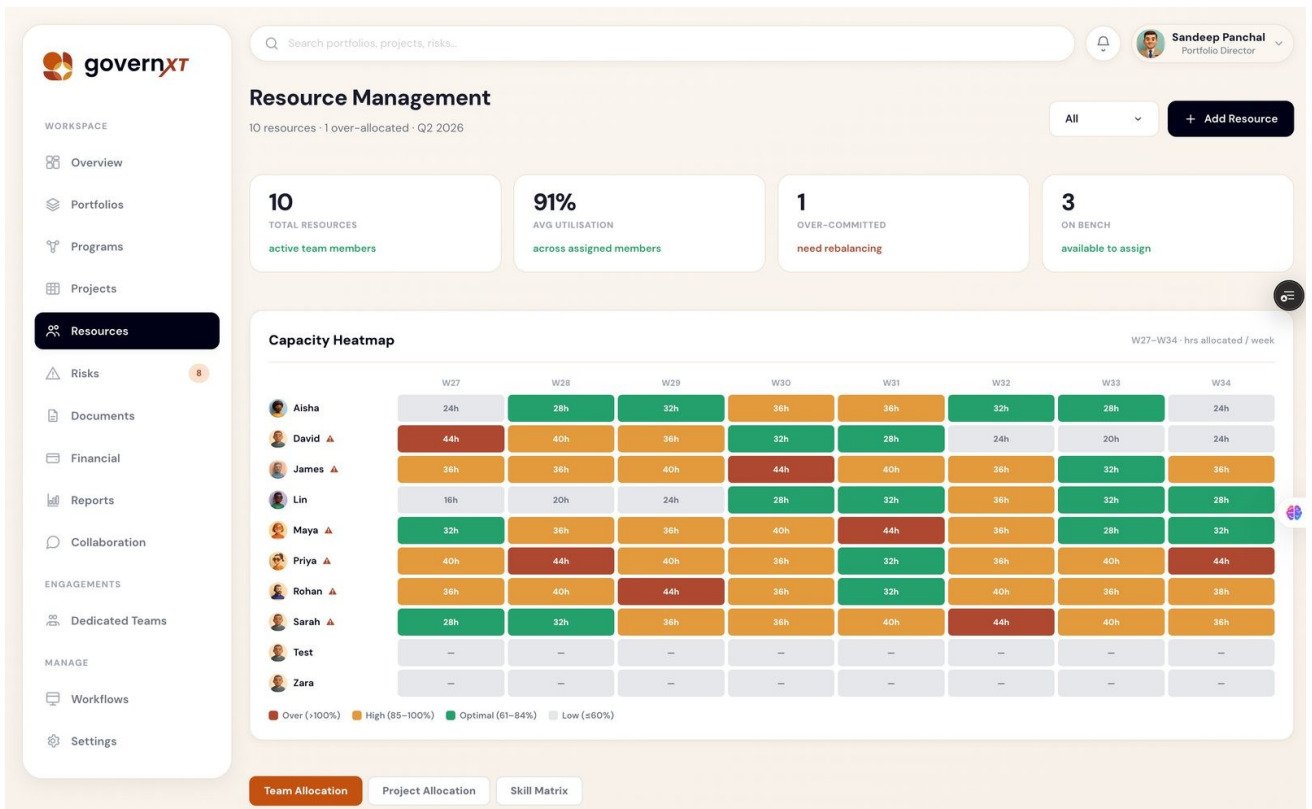


Figure 5 — Resource Management: eight-week capacity heatmap with over-allocation flags

Risk & Issue Management

The risk register scores every risk on probability × impact, tracks trend (improving, stable, worsening), owner and mitigation, and plots the whole register on a live 5×5 heat map. Three critical risks sit above the escalation threshold, each with a named mitigation in flight.

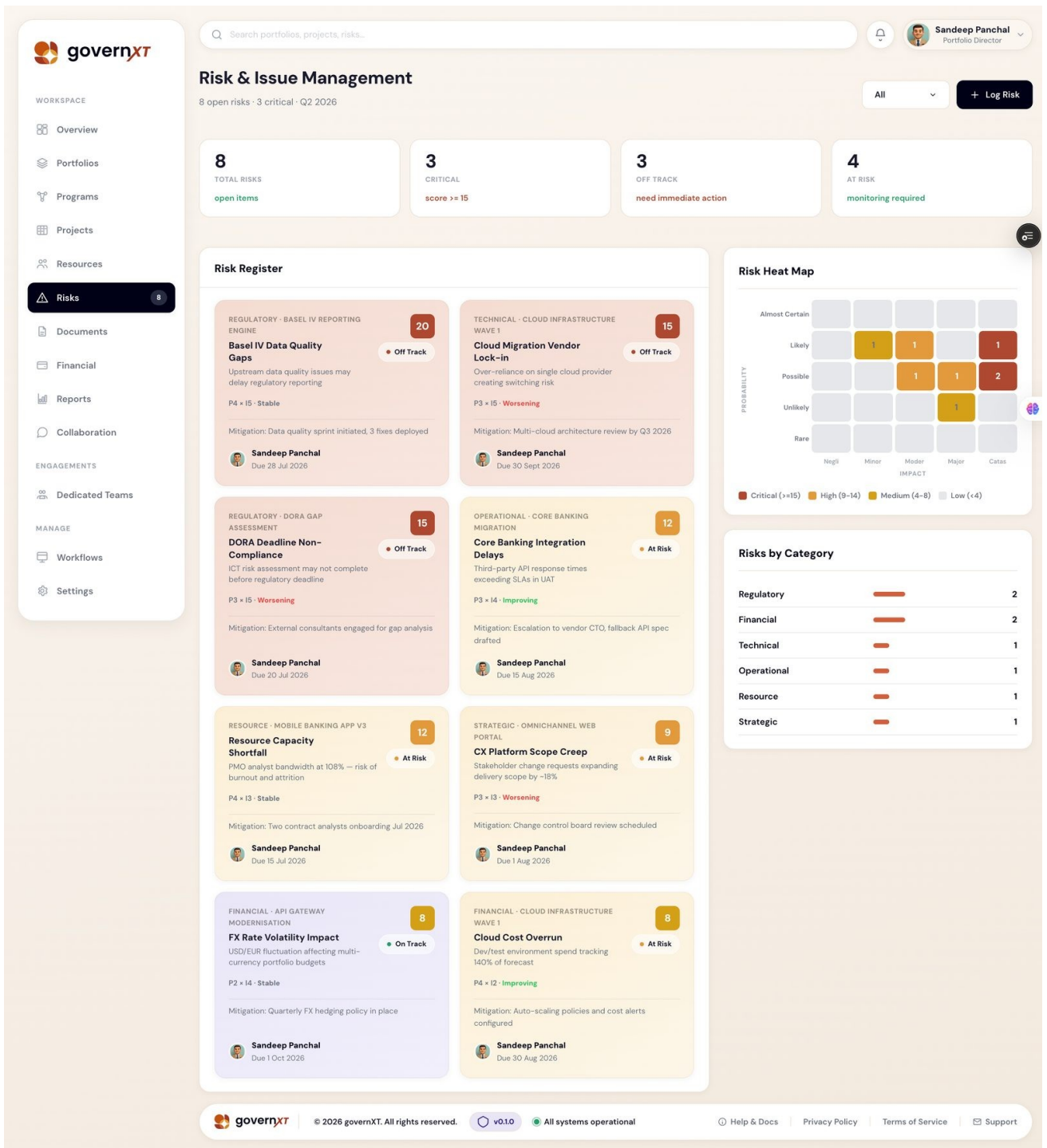


Figure 6 — Risk & Issue Management: scored register with live 5×5 probability/impact heat map

Financial Management

GovernXT implements textbook Earned Value Management: planned value, earned value and actual cost roll up across eleven projects into portfolio SPI and CPI, plotted on a twelve-month S-curve with a today marker and estimate-at-completion. A per-project EVM table exposes BAC, EAC, ETC and variance-at-completion.

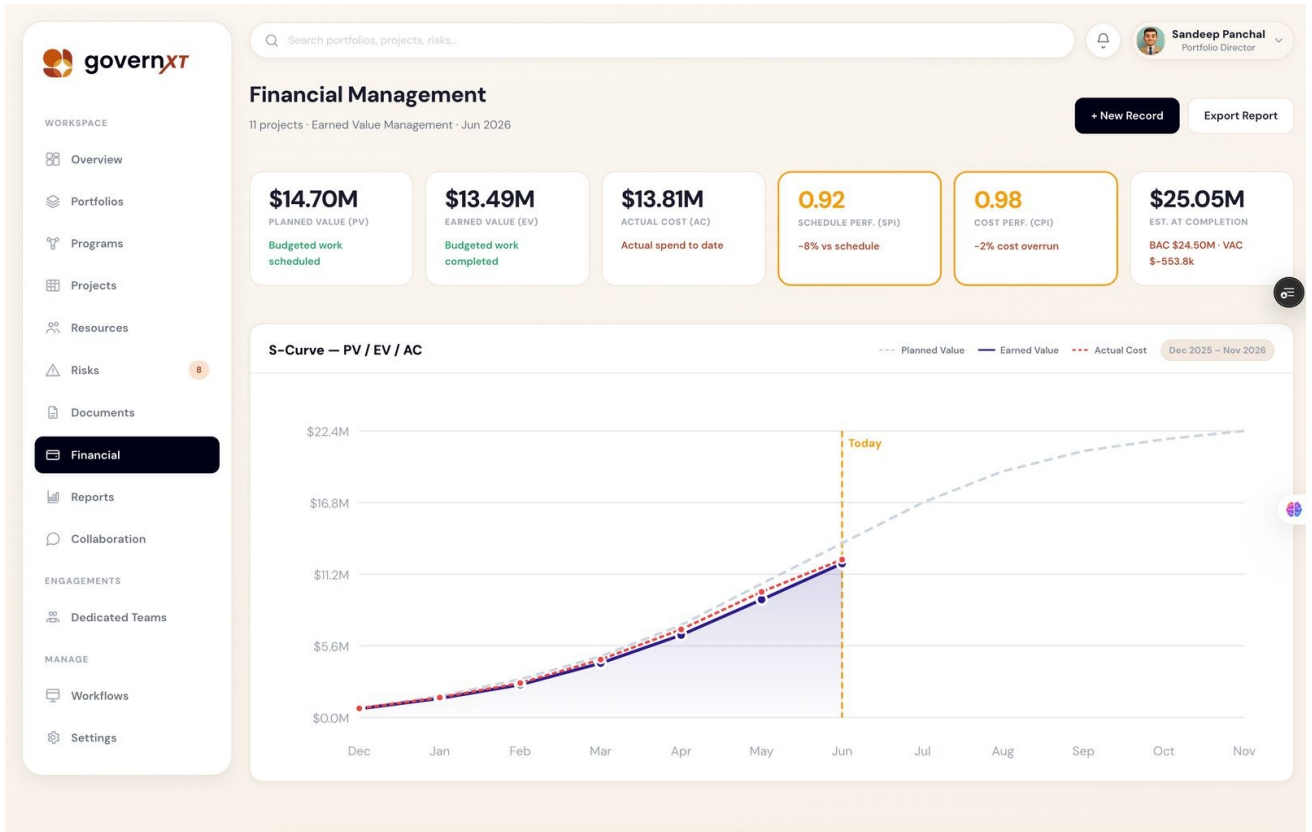


Figure 7 — Financial Management: portfolio-level EVM with PV/EV/AC S-curve

Change Management

Every scope, schedule or budget change flows through a governed approval chain — project manager, sponsor, PMO — rendered as a visual pipeline per request. Cost and schedule impact are first-class fields, so the module can report net cost impact across all change requests in one KPI.

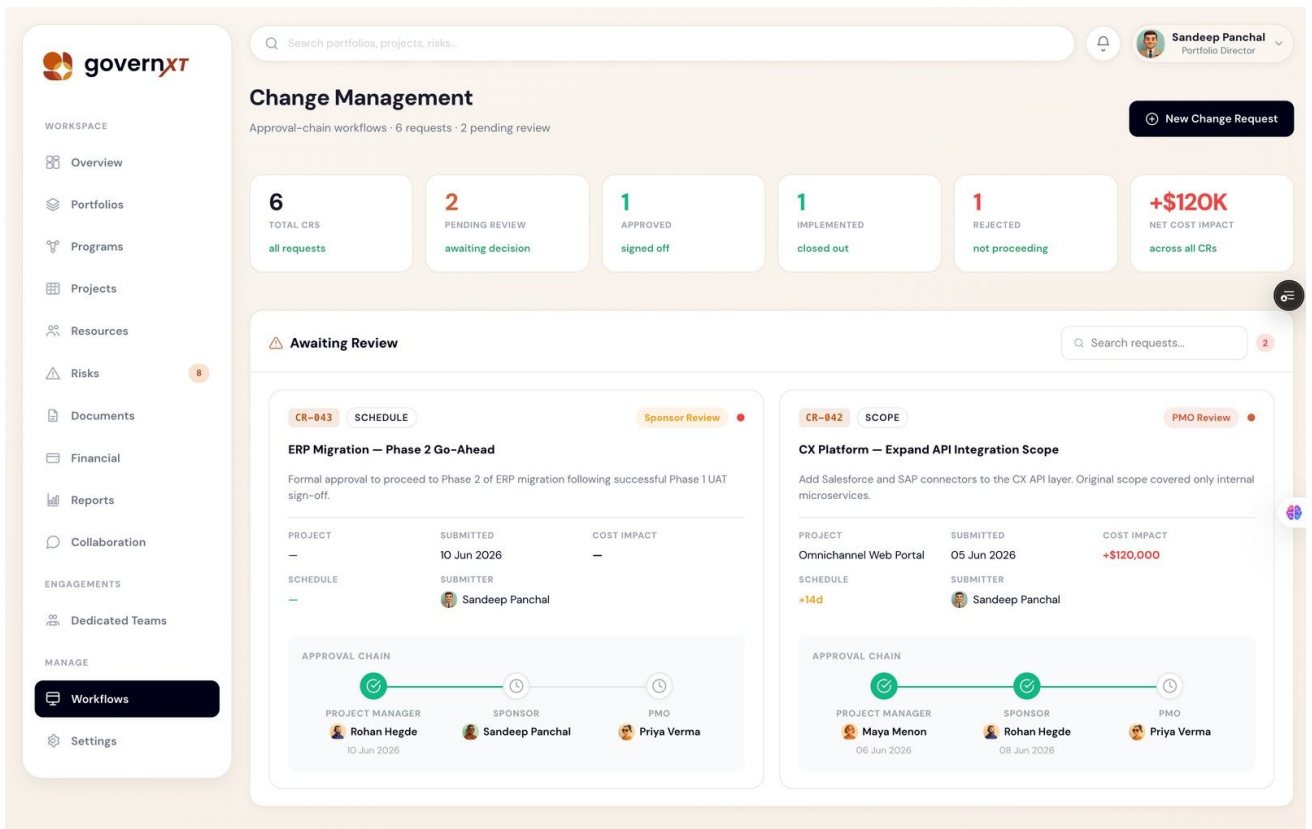


Figure 8 — Change Management: change requests moving through a three-step approval chain

Document Repository

Project documents are auto-versioned on re-upload and carry a status workflow from draft through review to final-and-locked. The register supports per-project, per-type and per-status filtering, tags, and full version history per document, backed by Supabase Storage with bucket-level access policies.

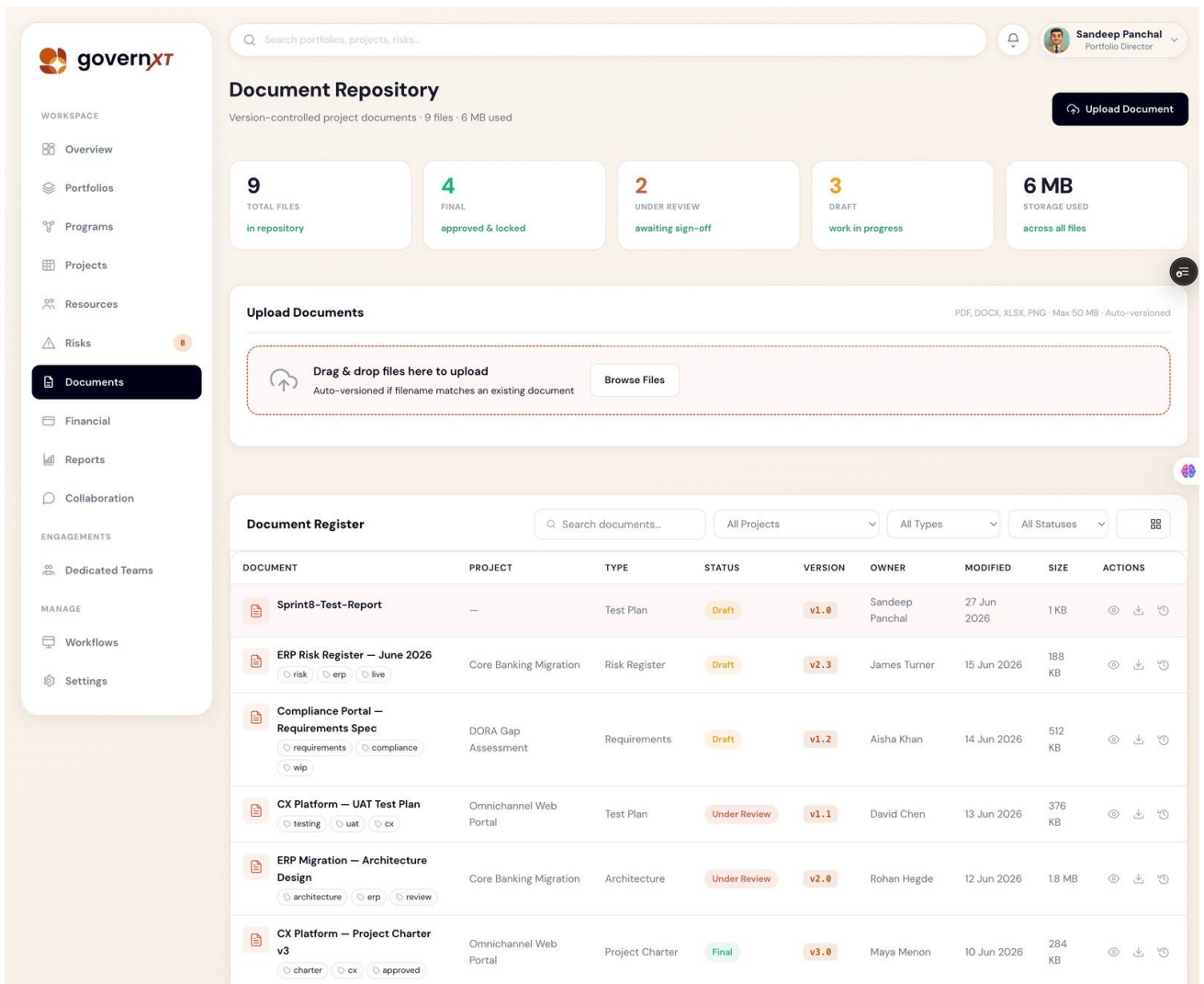


Figure 9 — Document Repository: version-controlled register with status workflow

Reports & Analytics

Thirteen pre-built reports span seven categories: live dashboards with real-time sparklines, scheduled weekly/monthly reports delivered as PDF or Excel, and on-demand runs. Sparklines are computed from live database series — project progress, risk scores, earned value and document volumes.

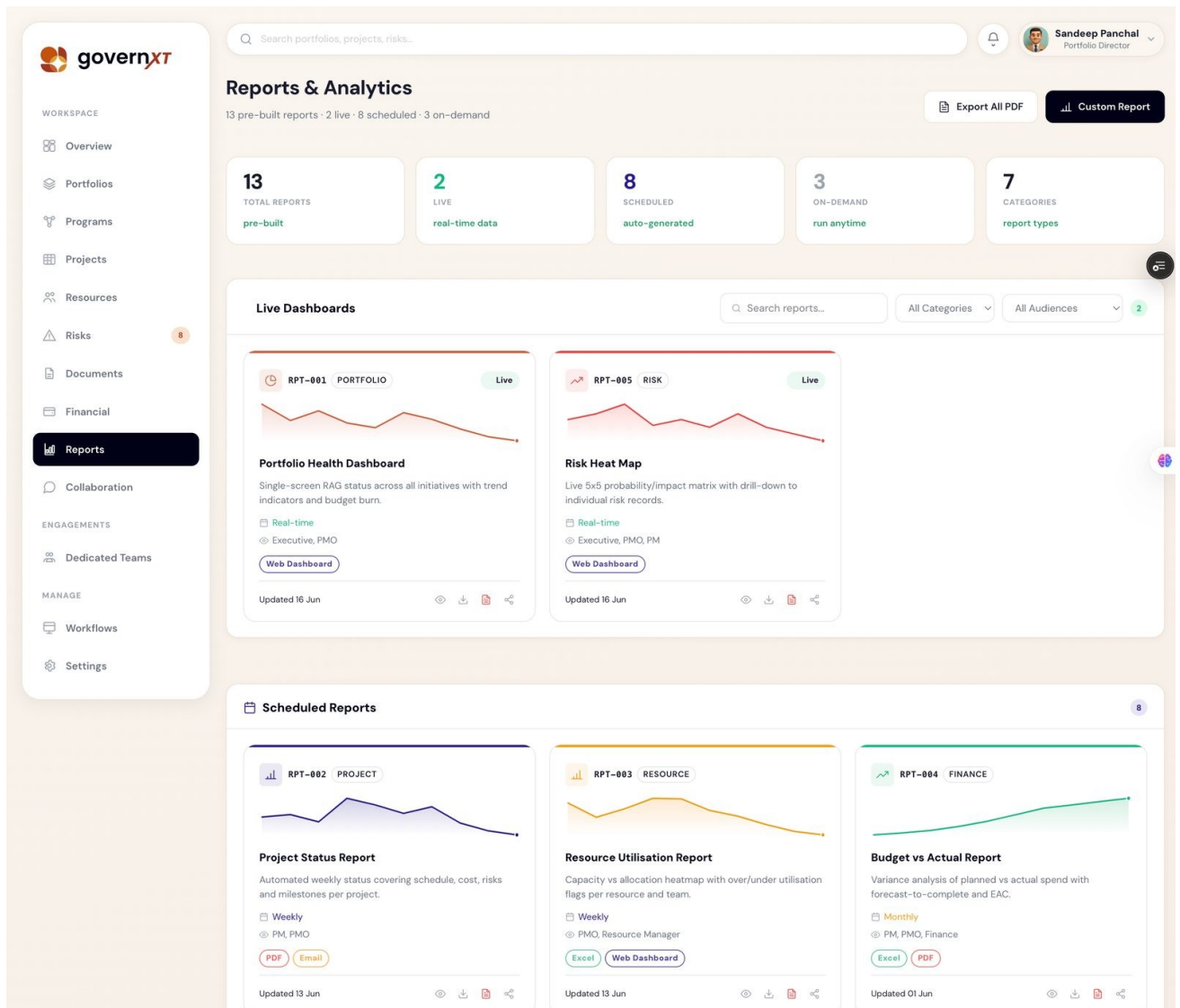


Figure 10 — Reports & Analytics: live dashboards, scheduled reports and on-demand exports

Collaboration

A unified activity feed captures approvals, risk escalations, document uploads and task movements across all projects, while threaded comments with @mentions and resolution states keep decisions attached to the work they concern.

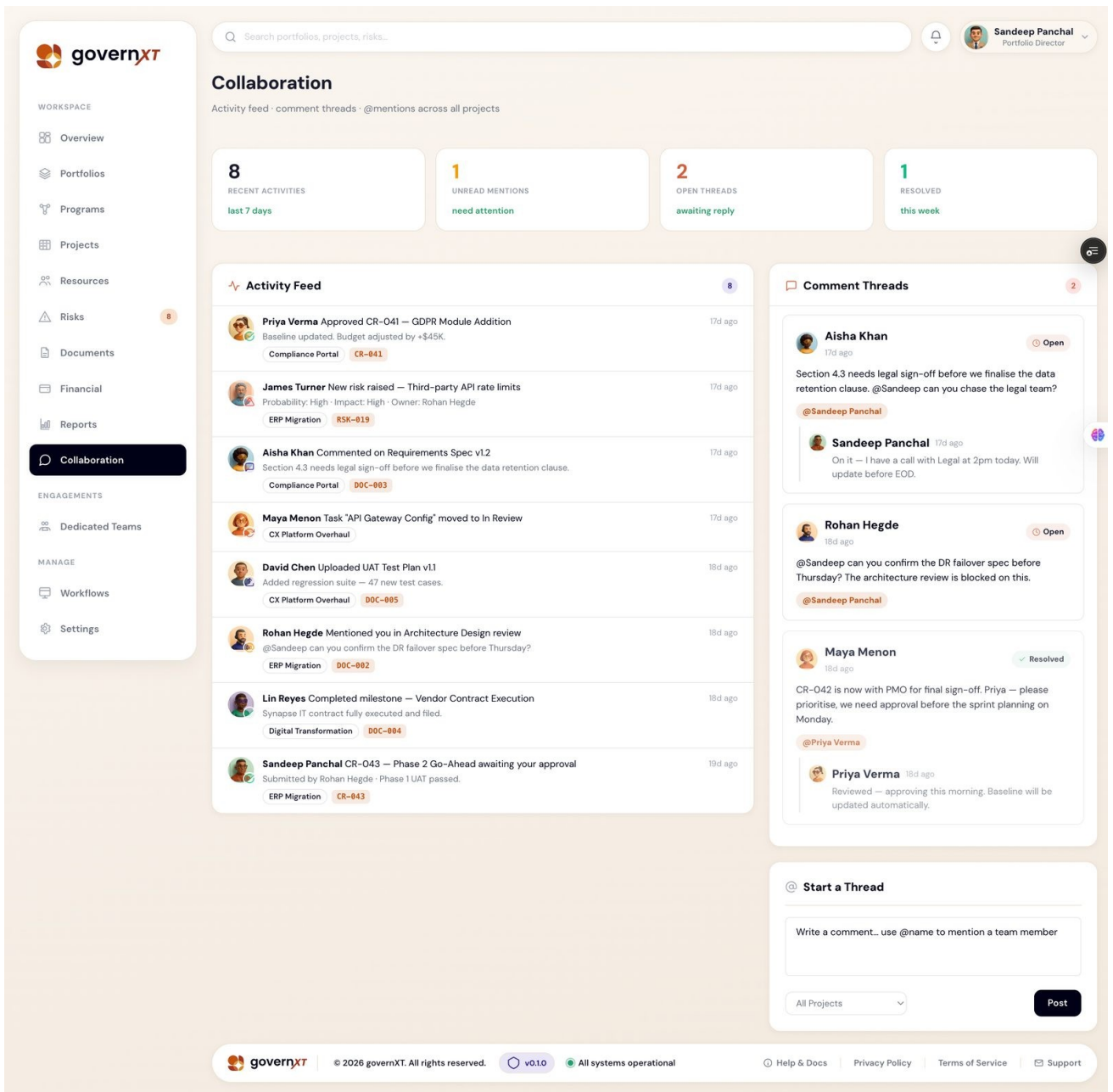


Figure 11 — Collaboration: cross-project activity feed with threaded @mention discussions

Admin & Governance

Organisation administrators manage users, roles, org settings and a full audit log from a four-tab console. Role assignment and deactivation are one click, and every administrative action lands in the audit trail.

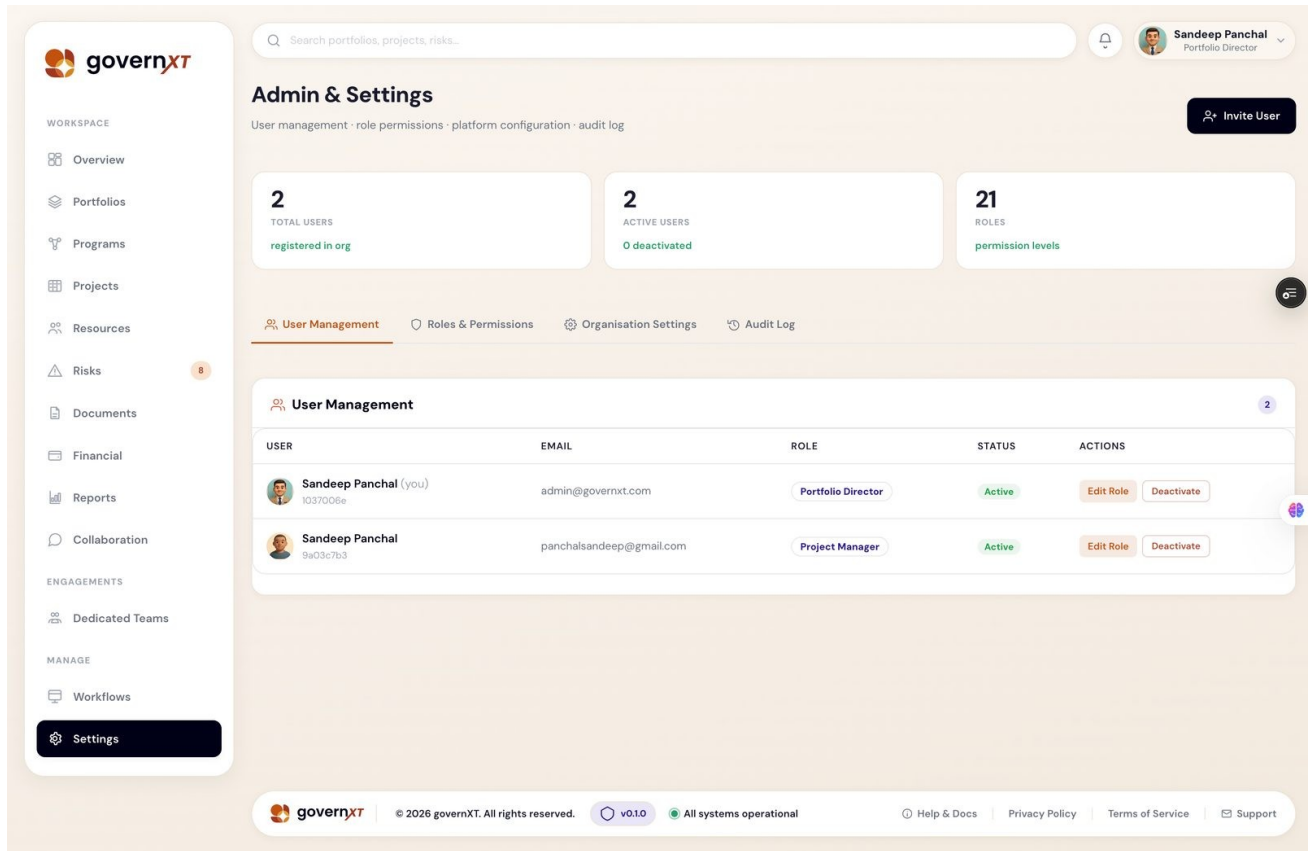


Figure 12 — Admin & Settings: user management with role assignment and deactivation

The platform ships seven fixed system roles that drive all permission logic, plus twenty-one renamable role labels so each organisation can speak its own language without touching security policy.

Roles & Permissions

Role codes are platform-managed and immutable — they anchor every RLS policy — while display labels are freely renamable per organisation.

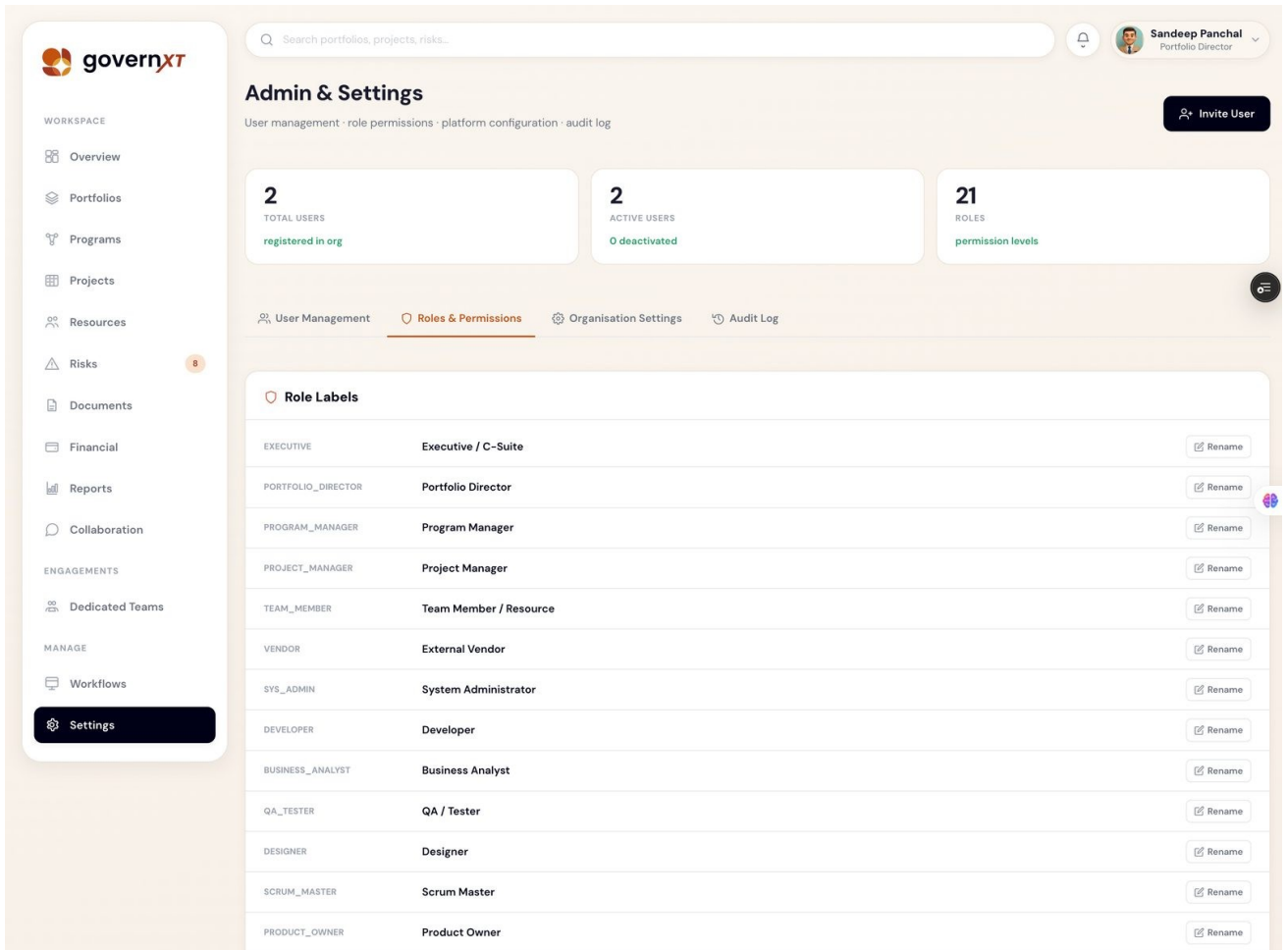


Figure 13 — Roles & Permissions: platform-managed role codes with per-organisation labels

Dedicated Teams

Beyond internal delivery, GovernXT models commercial engagements: dedicated teams with per-member day rates and allocations, margin and discount rules, and a live monthly pricing preview that recomputes base cost, gross price and profit as the team is edited.

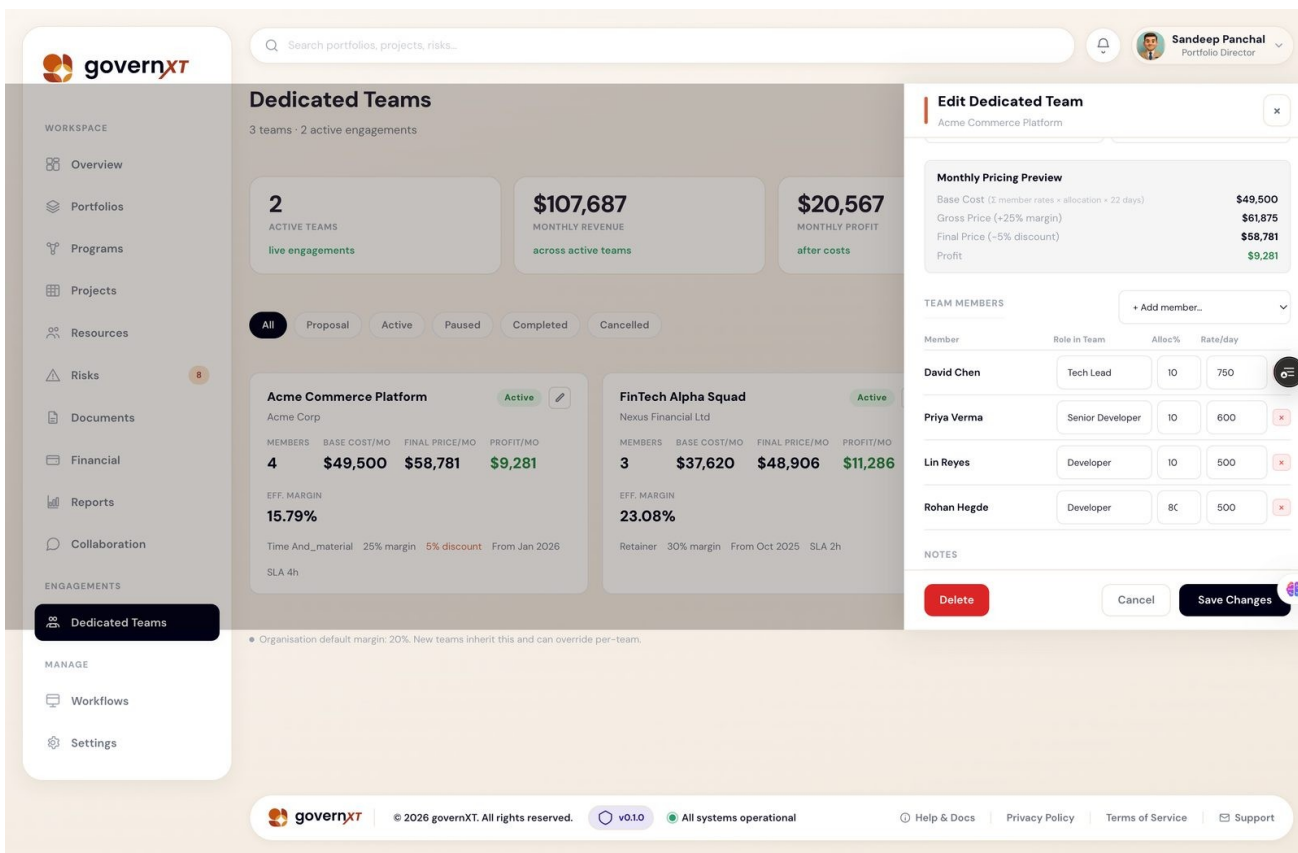


Figure 14 — Dedicated Teams: engagement economics with live pricing preview

Platform Administration (Multi-Tenant)

A separate, independently authenticated console governs the platform itself: tenant organisations, subscription plans, invitations and revenue. Plan distribution, users per organisation and MRR/ARR breakdowns give the operator a SaaS control room — cleanly separated from any tenant's project data.

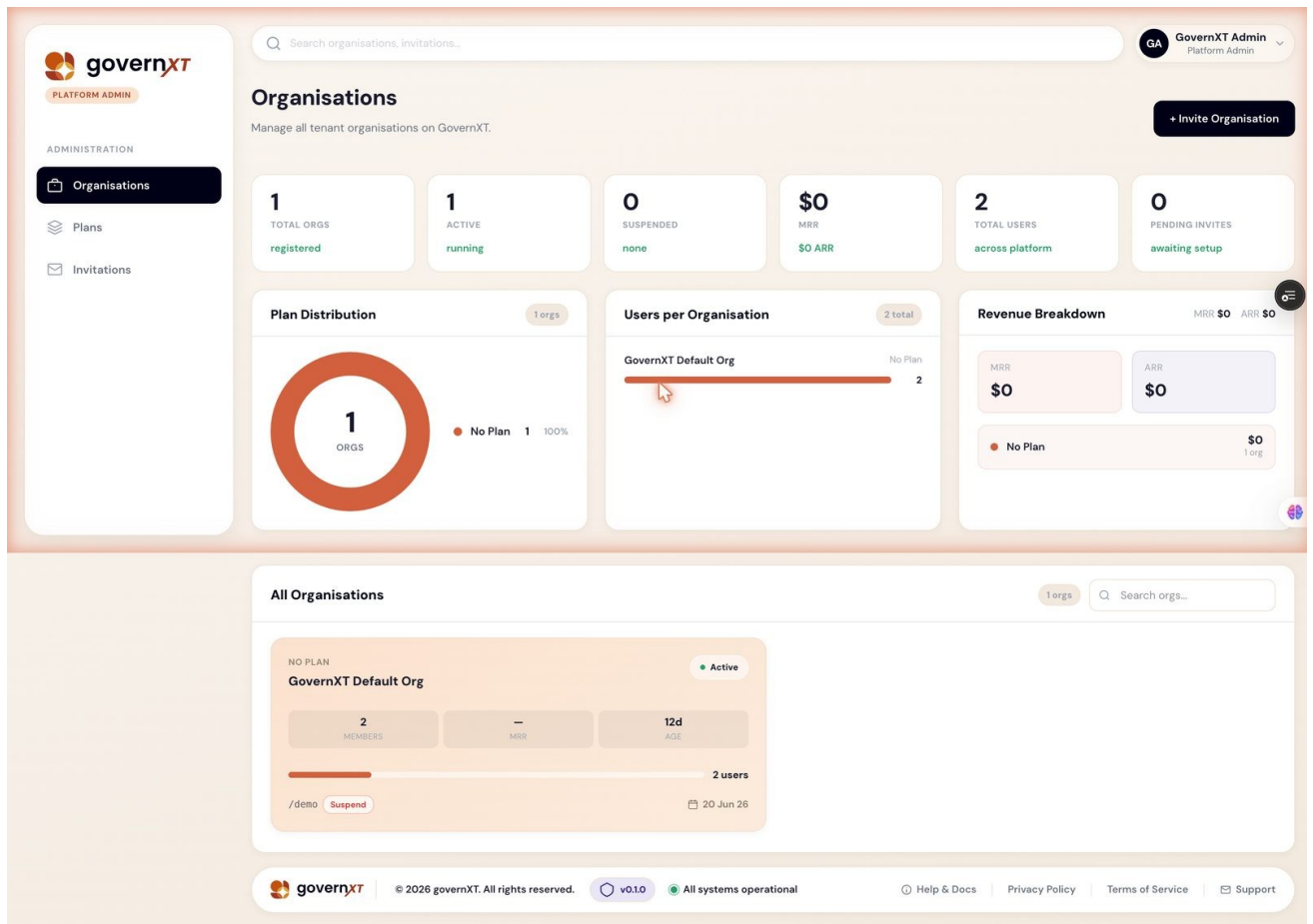


Figure 15 — Platform Admin: tenant organisations, plans and revenue breakdown

The Security Model

Seven platform-managed roles drive every Row-Level Security policy and permission check. Role codes are fixed; display labels are configurable per organisation.

| Role | Scope | Capabilities |
|------------------------|-------------------------------|--|
| Executive / C-Suite | Organisation-wide (read-only) | Portfolio health, ROI analytics, board-level approvals, executive reports |
| Portfolio Director | Organisation-wide (full) | Full CRUD on all modules, user management, approval chain setup, audit log |
| Program Manager | Assigned programs | Create and manage programs, program-level risks and resources |
| Project Manager | Assigned projects | Full CRUD on assigned projects, tasks, risks, CRs, documents, financials |
| Team Member / Resource | Assigned tasks | Update own tasks, submit timesheets, upload documents, comment |
| External Vendor | Configured per engagement | Restricted, admin-configured access that auto-expires |
| System Administrator | Platform admin only | Users, integrations, audit log, settings — no access to project data |

Because the same role model is enforced in the database, in middleware and in the UI, a compromised or buggy client cannot widen its own access: Postgres evaluates every row against the caller's organisation and role before returning it.

The Design System

GovernXT deliberately avoids stock component libraries. A bespoke design system — documented across eight versions and QA-audited on all sixteen routes — gives the product its signature warm, editorial look: cream canvas, ink typography and a terracotta accent family.

| | | | | |
|------------------------|-----------------------|------------------------------|-------------------------|------------------------|
| Navy #000016 | Ink #1A1A2E | Terracotta #D2603F | Amber #E29A3C | Rust #AE4830 |
| Primary CTAs | Typography | Brand accent, links | At-risk states | Off-track states |

- **Signature cards** — Warm gradient portfolio cards in terracotta, amber, rust and indigo are the signature element, used consistently across Portfolio, Programs, Resources and Risks.
- **Consistency** — A custom 24×24 icon set and a single CSS token system (dashboard.css) keep all sixteen routes visually identical — verified by a dedicated design-token QA pass.
- **Avatars** — Team avatars are AI-generated 3D characters (OpenAI gpt-image-1), stored in Supabase Storage and rendered through a fault-tolerant component with a three-stage fallback, so a missing image can never break the UI.

Engineering & Delivery Practice

The platform is delivered with the same governance discipline it sells:

- **Docs-first** — BRD, FRD, database structure, security layer and design system documents are versioned and updated before implementation — documents are the source of truth.
- **Sprint cadence** — A 54-user-story backlog is delivered in 12 planned sprints; sprints 1-4 (auth redirects, audit log, admin console, portfolio/program/project CRUD) are complete.
- **Verify-in-browser** — Every change passes a clean production build before push, deploys via Vercel, and is then verified in a live browser session with screenshots before being declared done.
- **Design QA pass** — All 16 routes were audited against a structural and token checklist; legacy CSS variables were eliminated from inline styles across the codebase.
- **Defensive data layer** — Ambiguous foreign-key joins use explicit constraint names, session checks avoid network round-trips, and destructive operations are blocked by database triggers rather than UI logic alone.

Current State & Roadmap

All sixteen dashboard routes are live against Supabase with full read models, task management (Kanban, Gantt, WBS) is interactive, and the multi-tenant platform console is operational. The active roadmap completes write operations across the remaining modules — resource allocation, financial data entry, report builder, notifications and global search — over the remaining sprint plan through February 2027.

About the Builder

Sandeep Panchal designed, engineered and operates GovernXT end-to-end: product definition, data modelling, security architecture, UI design system, frontend and backend implementation, deployment and QA. The project demonstrates that with modern serverless infrastructure and disciplined engineering practice, a single builder can ship — and govern — an enterprise-grade platform.

Sandeep Panchal

Platform Lead — GovernXT

www.governxt.com · panchalsandeep@gmail.com